

DWEC – Javascript Web Cliente.

DWEC – Páginas html

En una aplicación web basada en Node.js, Express y MySQL, generalmente no suele haber un único archivo HTML. Por lo general, las aplicaciones web tienen múltiples páginas o vistas, y cada una de ellas tiene su propio archivo HTML. Estos archivos HTML pueden estar organizados en un directorio específico, como por ejemplo un directorio llamado "views".

En una aplicación web típica, las rutas definidas en Express pueden renderizar diferentes vistas según las solicitudes del usuario. Puedes usar un motor de plantillas (template engine) como EJS, Pug, o Handlebars para generar HTML dinámicamente en el servidor y luego enviarlo al cliente.

Un ejemplo sencillo con Express y EJS podría verse así:

1. Organización de carpetas:


markdown

Copy code

```
- proyecto
  - public
    - styles
      - style.css
  - views
    - index.ejs
    - otraVista.ejs
  - node_modules
  - app.js
  - package.json
```

1. Archivo `app.js`:

javascript

 Copy code

```
const express = require('express');
const app = express();
const port = 3000;

app.set('view engine', 'ejs');

app.get('/', (req, res) => {
  res.render('index');
});

app.get('/otraRuta', (req, res) => {
  res.render('otraVista');
});

app.listen(port, () => {
  console.log(`La aplicación está escuchando en http://localhost:${port}`);
});
```

1. Archivo `views/index.ejs`:

```
html Copy code

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Mi Aplicación</title>
  <link rel="stylesheet" href="/styles/style.css">
</head>
<body>
  <h1>Bienvenido a mi aplicación</h1>
  <!-- Contenido de la página -->
</body>
</html>
```

Cada ruta en tu aplicación puede renderizar una vista diferente. La estructura exacta puede variar según las necesidades específicas de tu aplicación, pero generalmente se prefiere organizar los archivos HTML en directorios específicos para facilitar la administración del código fuente.

Página de entrada a la aplicación

La página de entrada o página principal de una aplicación web suele ser determinada por la ruta definida en Express. En el ejemplo anterior, la página de entrada se establece en la ruta raíz ("/") mediante la siguiente declaración:

```
javascript Copy code

app.get('/', (req, res) => {
  res.render('index');
});
```

En este caso, cuando un usuario accede a la URL principal de la aplicación (por ejemplo, `http://localhost:3000/`), se renderiza la vista llamada `'index.ejs'`. Esta vista representa la página principal de la aplicación.

Sin embargo, ten en cuenta que la designación de la página de entrada puede variar según la estructura y la lógica de tu aplicación. Puedes configurar Express para que la página de entrada sea diferente o incluso dinámica según ciertas condiciones o la autenticación del usuario. La página de entrada es simplemente la primera página que se muestra al usuario cuando accede al sitio web.

Enfoque SPA (Aplicación de página única)

Las "aplicaciones de página única" (SPA, por sus siglas en inglés), se refieren a un enfoque de desarrollo web en el que la interfaz de usuario se carga y actualiza dinámicamente en una sola página, en lugar de cargar páginas HTML completamente nuevas desde el servidor. Este enfoque proporciona una experiencia de usuario más fluida, ya que solo se cargan los datos necesarios y no toda la página.

Las SPAs suelen ser construidas utilizando frameworks o bibliotecas de JavaScript, como React, Angular o Vue.js. Estos frameworks permiten la creación de componentes reutilizables que se pueden actualizar de manera eficiente en respuesta a las interacciones del usuario sin recargar la página completa.

En el contexto de una SPA, la "página de entrada" se refiere a la primera vista que se carga cuando el usuario accede al sitio web. Esta vista puede incluir la estructura principal de la aplicación y cargar dinámicamente los datos necesarios. Sin embargo, en una SPA, el término "página de entrada" puede ser un poco menos relevante que en las aplicaciones tradicionales de varias páginas.

La elección entre el desarrollo de SPA y aplicaciones tradicionales de varias páginas depende de los requisitos específicos del proyecto y las preferencias del equipo de desarrollo. Ambos enfoques tienen sus ventajas y desventajas, y la elección dependerá de factores como la complejidad de la aplicación, la experiencia del usuario deseada y los recursos disponibles.